UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/643,574	08/18/2003	James R. Kohn	1376.730US1	3940

21186 7590 01/30/2006

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH
1600 TCF TOWER
121 SOUTH EIGHT STREET
MINNEAPOLIS, MN 55402

EXAMINER	
FENNEMA, ROBERT E	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 01/30/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/643,574	KOHN, JAMES R.	
	Examiner	Art Unit	
	Robert E. Fennema	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 August 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-34* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *18 August 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-34 are pending.

Specification

2. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Drawings

3. New corrected drawings in compliance with 37 CFR 1.121(d) are required in this application because the submitted drawings are informal. Applicant is advised to employ the services of a competent patent draftsperson outside the Office, as the U.S. Patent and Trademark Office no longer prepares new drawings. The corrected drawings are required in reply to the Office action to avoid abandonment of the application. The requirement for corrected drawings will not be held in abeyance.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 5-6, 16-17, and 25-26 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one

skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. The Claims stated above all state that a sequence of values are formed by concatenating a portion of each respective addressing value of the first vector of addressing values to a respective one of a sequence of numbers. It is believed by the Examiner that the sequence of values is important to the functioning of the invention, yet there is no disclosure of what the sequence of numbers can or has to be, or what their intended purpose is.

6. Claim 8 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claim 8 discloses that the operations "recited therein" are executed in "the order recited therein". A specific order must be given; the current wording is too vague to enable one of ordinary skill in the art to correctly interpret the necessary ordering of the steps.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-10, and 20-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beard et al. (USPN 5,640,524, herein Beard), in view of Bruckert et al. (USPN 5,068,851, herein Bruckert), and further in view of Ernst et al. (herein Ernst).

9. As per Claim 1, Beard teaches: A computerized method comprising:

providing a first vector of addressing values (Column 2 Line 65 – Column 3 Line 1);

providing a second vector of operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

storing a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each location's address is based at least in part on a corresponding one of the addressing values (Column 3, Lines 19-22);

reading back from the sequence of addressed locations values resulting from the storing of the first sequence to obtain a second sequence of values (Column 3 Lines 5-10);

performing an arithmetic-logical operation using values from the third vector register and the compressed second vector of operand values to generate a result vector (Column 3, Lines 10-13); and

using the first vector of addressing values as masked by the bit vector, storing the result vector to memory (Column 3, Lines 11-13), but fails to teach:

comparing the first sequence of values to the second sequence of values to generate a bit vector representing compares and miscompares;

compressing the second vector of operand values using the bit vector;

using the first vector of addressing values as masked by the bit vector, loading a third vector register with elements from memory.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention. Ernst teaches compressing the second vector by marking certain instructions as invalid (Section 2.4, by marking the instruction invalid, it is compressing in the sense that the vector contains less useful information), and using those values, loading a third vector register with elements from memory (Section 2.4, the invalid instructions must replay, which when combined with Beards invention, would require those particular instructions to be read out into another vector correctly). One of ordinary skill in the art would recognize the advantage in selectively reloading/replaying is that only a few instructions must be re-executed, as opposed to the entire block of instructions. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

10. As per Claim 2, Beard teaches: The method of claim 1, wherein addresses of the elements in memory are calculated by adding each respective addressing value to a base address of an object in memory (Column 3 Lines 2-5).

11. As per Claim 3, Beard teaches: The method of claim 1, wherein the arithmetic-logical operation is an addition operation that produces at least one element of the result vector as a summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values (it is well known in the art that a single vector/array value can be added to multiple elements of another vector/array. For example, see Cohoon et al., Page 493, Program 10.2) corresponding to elements of the first vector of addressing values that had identical values (In correcting errors in a vector write, it would not make sense to operate on values that were already correct, therefore only the incorrect (identical/duplicate) values should be modified).

12. As per Claim 4, Beard teaches: The method of claim 1, wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated using a truncated portion of each respective addressing value of the first vector of addressing values (Column 7 Lines 30-36. Bits 6-0 (which are truncated from the 32-bit value) are used to index into memory).

13. As per Claim 5, Beard teaches: The method of claim 4, wherein data values of the first sequence of values are each formed by concatenating a portion of each respective addressing value of the first vector of addressing values to a respective one of a sequence of numbers (It is a common practice in computing to "sign extend"

values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12).

14. As per Claim 6, Beard teaches: The method of claim 1 wherein the constrained area of memory includes 2^N locations, wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated by adding a base address to an N-bit portion of each respective addressing value of the first vector of addressing values (Column 3 Lines 2-5 discloses adding a value to a base address, and it is well known in the art/inherent that if you have an N-bit address, it can address an area of memory of size 2^N), and wherein data values of the first sequence of values are each formed by concatenating a portion of each respective addressing value of the first vector of addressing values to a respective one of a consecutive sequence of integer numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12).

15. As per Claim 7, Ernst teaches: The method of claim 1, wherein for the loading of the third vector register with elements from memory, elements are loaded from locations specified by addressing values corresponding to bits of the bit vector that indicated a compare and no elements are loaded from locations specified by addressing values corresponding to bits of the bit vector that indicated a miscompare (Section 2.4, when selectively reloading the instructions to be replayed, only the ones that matched the criteria (i.e., not ready/valid), would have to be loaded).

16. As per Claim 8, Beard teaches: The method of claim 1, wherein the operations recited therein are executed in the order recited therein (Inherent that they are executed in the order stated, otherwise nonsense output would be generated, as each step relies on the previous one to execute properly).

17. As per Claim 9, Beard teaches: The method of claim 1, further comprising:

performing a first synchronization operation that ensures that the comparing the first sequence of values to the second sequence of values to generate the bit vector representing compares and miscompares effectively completes before the loading of the third vector register with elements from memory (inherent that Beards machine must compare before loading the third vector register, as loading the vector register is dependant upon the compare results, as seen in Claim 1 rejection); and

performing a first synchronization operation that ensures that the storing the result vector to memory completes before subsequent passes through a loop (inherent

that in a non-bypassing, in-order processor, that all results must commit before a depending instruction can run, to ensure consistency. There is no evidence in the specification that the method is designed to run speculatively or out of order, and as such cannot execute instructions before a previous block is complete. See Patterson et al., Pages 195-196 for further information on why in-order retirement (and without speculation, in-order execution) is required in modern processors).

18. As per Claim 10, Beard teaches: A computer-readable medium having instructions stored thereon for causing a suitably programmed information-processing system to execute a method comprising:

providing a first vector of addressing values (Column 2 Line 65 – Column 3 Line 1);

providing a second vector of operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

storing a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each location's address is based at least in part on a corresponding one of the addressing values (Column 3, Lines 19-22);

reading back from the sequence of addressed locations values resulting from the storing of the first sequence to obtain a second sequence of values (Column 3 Lines 5-10);

performing an arithmetic-logical operation using values from the third vector register and the compressed second vector of operand values to generate a result vector (Column 3, Lines 10-13); and

using the first vector of addressing values as masked by the bit vector, storing the result vector to memory (Column 3, Lines 11-13), but fails to teach:

comparing the first sequence of values to the second sequence of values to generate a bit vector representing compares and miscompares;

compressing the second vector of operand values using the bit vector;

using the first vector of addressing values as masked by the bit vector, loading a third vector register with elements from memory.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention. Ernst teaches compressing the second vector by marking certain instructions as invalid (Section 2.4, by marking the instruction invalid, it is compressing in the sense that the vector contains less useful information), and using those values, loading a third vector register with elements from memory (Section 2.4, the invalid instructions must replay, which when combined with Beards invention, would require those particular instructions to be read out into another vector correctly). One of ordinary skill in the art would

recognize the advantage in selectively reloading/replaying is that only a few instructions must be re-executed, as opposed to the entire block of instructions. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

19. As per Claim 20, Beard teaches: The method of claim 11,

wherein the constrained area contains 2^N consecutive addresses (Column 3 Lines 2-5 discloses adding a value to a base address, and it is well known in the art/inherent that if you have an N-bit address, it can address an area of memory of size 2^N),

wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated using an N-bit value derived from each respective addressing value of the first vector register of addressing values (Column 7 Lines 30-36),

wherein data values of the first sequence of values are each formed by combining at least a portion of each respective addressing value of the first vector register of addressing values to a respective one of a consecutive sequence of integer numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the

concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12),

wherein addresses of the elements from memory are calculated by adding each respective addressing value to a base address (Column 3 Lines 2-5),

wherein the arithmetic-logical operation is a floating-point addition operation that produces at least one element of the result vector as an ordered-operation floating point summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector register of addressing values having identical values (Column 4 Line 65 – Column 5 Line 19 shows floating point to be one of the available operations. Also see Column 3, Lines 10-13), but fails to teach:

wherein for the loading of the third vector register with elements from memory, elements are loaded from locations specified by addressing values corresponding to indications that indicated compares and no elements are loaded from locations specified by addressing values corresponding to indications that indicated miscompares,

wherein for the storing of the result vector of elements to memory, elements are stored to locations specified by addressing values corresponding to indications that indicated compares and no elements are stored to locations specified by addressing values corresponding to indications that indicated miscompares.

Ernst teaches selectively replaying instructions that have an invalid bit set (Section 2.4). One of ordinary skill in the art would recognize that selectively replaying

instructions means that you would only operate on values that were specified as needing to be replayed, IE, those locations that had a positive compare. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

20. As per Claim 21, Beard teaches: A system comprising:

a first vector processor having:

a first vector register having addressing values (Column 2 Line 65 – Column 3 Line 1);

a second vector register having operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

a third vector register (Column 2, Lines 62-63, there are a plurality of registers);

a bit vector register (Column 2, Lines 62-63, a bit vector register is a register that holds more than 1 bit, which by definition would be a vector register);

circuitry that selectively stores a first sequence of values to a sequence of addressed locations within a constrained area of memory (Column 3, Lines 19-22),

wherein each location's address is based at least in part on a corresponding one of the addressing values (Column 3, Lines 19-22);

circuitry that selectively loads, from the sequence of addressed locations, values resulting from the stores of the first sequence to obtain a second sequence of values (Column 3 Lines 5-10);

circuitry that selectively performs an arithmetic-logical operation on corresponding values from the third vector register and the compressed second vector of operand values to generate values of a result vector (Column 3, Lines 10-13); and;

circuitry that selectively stores the result vector to memory (Column 3, Lines 11-13), but fails to teach:

circuitry that selectively compares the first sequence of values to the second sequence of values to generate bit values into the bit vector register representing compares and miscompares;

circuitry that selectively compresses the second vector of operand values using the values in the bit vector register;

circuitry that selectively loads the third vector register with elements from memory addresses generated from the first vector register of addressing values as masked by the bit vector register.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention. Ernst teaches compressing the second vector by marking certain instructions as invalid (Section 2.4, by marking the instruction invalid, it is compressing in the sense that the vector contains less useful information), and using those values, loading a third vector

register with elements from memory (Section 2.4, the invalid instructions must replay, which when combined with Beards invention, would require those particular instructions to be read out into another vector correctly). One of ordinary skill in the art would recognize the advantage in selectively reloading/replaying is that only a few instructions must be re-executed, as opposed to the entire block of instructions. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

21. As per Claim 22, Beard teaches: The system of claim 21, further comprising circuitry to calculate addresses of the elements in memory by adding each respective addressing value to a base address value (Column 3 Lines 2-5).

22. As per Claim 23, Beard teaches: The system of claim 21, wherein the arithmetic-logical operation is an addition operation that produces at least one element of the result vector as a summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values (it is well known in the art that a single vector/array value can be added to multiple elements of another vector/array. For example, see Cohoon et al., Page 493, Program 10.2) corresponding to elements of the first vector register of addressing values that had identical values (In correcting errors in a vector write, it would not make sense to operate on values that were already correct, therefore only the incorrect (identical/duplicate) values should be modified).

23. As per Claim 24, Beard teaches: The system of claim 21, further comprising circuitry to calculate address values for the sequence of addressed locations within the constrained area of memory using a truncated portion of each respective addressing value of the first vector register of addressing values (Column 7 Lines 30-36. Bits 6-0 (which are truncated from the 32-bit value) are used to index into memory).

24. As per Claim 25, Beard teaches: The system of claim 24, further comprising circuitry to generate data values of the first sequence of values by joining a portion of each respective addressing value of the first vector register of addressing values to a respective one of a sequence of numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12).

25. As per Claim 26, Beard teaches: The system of claim 21, further comprising circuitry to generate address values of the sequence of addressed locations within the constrained area of memory by adding a base address to an N-bit portion of each respective addressing value of the first vector register of addressing values (Column 3 Lines 2-5 discloses adding a value to a base address, and it is well known in

the art/inherent that if you have an N-bit address, it can address an area of memory of size 2^N); and

circuitry to generate data values of the first sequence of values by combining a portion of each respective addressing value of the first vector register of addressing values with a respective one of a consecutive sequence of integer numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12).

26. As per Claim 27, Ernst teaches: The system of claim 21, wherein the circuitry that selectively loads the third vector register with elements from memory only loads element from locations specified by addressing values corresponding to bits of the bit vector that indicated a compare (Section 2.4, when selectively reloading the instructions to be replayed, only the ones that matched the criteria (i.e., not ready/valid), would have to be loaded).

27. As per Claim 28, Beard teaches: The system of claim 21, further comprising:

synchronization circuitry that ensures that the comparing the first sequence of values to the second sequence of values to generate the bit vector representing compares and miscompares effectively completes before the loading of the third vector

register with elements from memory (inherent that Beards machine must compare before loading the third vector register, as loading the vector register is dependant upon the compare results, as seen in Claim 1 rejection), and that ensures that the storing the result vector to memory completes before subsequent passes through a loop (inherent that in a non-bypassing, in-order processor, that all results must commit before a depending instruction can run, to ensure consistency. There is no evidence in the specification that the method is designed to run speculatively or out of order, and as such cannot execute instructions before a previous block is complete. See Patterson et al., Pages 195-196 for further information on why in-order retirement (and without speculation, in-order execution) is required in modern processors).

28. As per Claim 29, Beard teaches: The system of claim 21, further comprising:

a second vector processor having (Column 1, Lines 48-50, which states this can be a multiprocessor system):

a first vector register having addressing values (Column 2 Line 65 – Column 3 Line 1);

a second vector register having operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

a third vector register (Column 2, Lines 62-63, there are a plurality of registers);

a bit vector register (Column 2, Lines 62-63, there are a plurality of registers);

circuitry that selectively stores a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each location's address is based at least in part on a corresponding one of the addressing values (Column 3, Lines 19-22);

circuitry that selectively loads, from the sequence of addressed locations, values resulting from the stores of the first sequence to obtain a second sequence of values (Column 3 Lines 5-10);

circuitry that selectively performs an arithmetic-logical operation on corresponding values from the third vector register and the compressed second vector of operand values to generate values of a result vector (Column 3 Lines 10-13); and;

circuitry that selectively stores the result vector to memory (Column 3, Lines 11-13); and

synchronization circuitry that ensures that the comparing the first sequence of values to the second sequence of values to generate the bit vector representing compares and miscompares effectively completes in both the first and second vector processors before the loading of the third vector register with elements from memory in either processor (inherent that Beards machine must compare before loading the third vector register, as loading the vector register is dependant upon the compare results, as seen in Claim 21 rejection), and that ensures that the storing the result vector to memory completes before subsequent passes through a loop (inherent that in a non-bypassing, in-order processor, that all results must commit before a depending instruction can run, to ensure consistency. There is no evidence in the specification that

the method is designed to run speculatively or out of order, and as such cannot execute instructions before a previous block is complete. See Patterson et al., Pages 195-196 for further information on why in-order retirement (and without speculation, in-order execution) is required in modern processors), but fails to teach:

circuitry that selectively compares the first sequence of values to the second sequence of values to generate bit values into the bit vector register representing compares and miscompares;

circuitry that selectively compresses the second vector of operand values using the values in the bit vector register;

circuitry that selectively loads the third vector register with elements from memory addresses generated from the first vector register of addressing values as masked by the bit vector register.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention. Ernst teaches compressing the second vector by marking certain instructions as invalid (Section 2.4, by marking the instruction invalid, it is compressing in the sense that the vector contains less useful information), and using those values, loading a third vector register with elements from memory (Section 2.4, the invalid instructions must replay,

which when combined with Beards invention, would require those particular instructions to be read out into another vector correctly). One of ordinary skill in the art would recognize the advantage in selectively reloading/replaying is that only a few instructions must be re-executed, as opposed to the entire block of instructions. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

29. Claims 11-20 and 30-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beard, in view of Brucket.

30. As per Claim 11, Beard teaches: A computerized method comprising:

loading a first vector register with addressing values (Column 2 Line 65 – Column 3 Line 1);

loading a second vector register with operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

storing a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each one of these location's addresses in the constrained area of memory is based at least in part on a subset of bits of a corresponding one of the addressing values (Column 3, Lines 19-22);

reading back from the sequence of addressed locations values resulting from the storing of the first sequence to obtain a second sequence of values (Column 3 Lines 5-10);

selectively combining, with an arithmetic-logical operation, certain elements of the second vector of operand values based on results of the comparing (Column 3, Lines 10-13);

performing the arithmetic-logical operation using values from the third vector register and the combined second vector of operand values to generate a result vector (Column 3, Lines 10-13); and

using the at least some of the first vector register of addressing values, storing the result vector to memory (Column 3, Lines 11-13), but fails to teach:

comparing the first sequence of values to the second sequence of values;

using at least some of the first vector register of addressing values, loading a third vector register with elements from memory.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

31. As per Claim 12, Beard teaches: The method of claim 11, wherein addresses of the elements from memory are calculated by adding each respective addressing value to a base address (Column 3 Lines 2-5).

32. As per Claim 13, Beard teaches: The method of claim 11, wherein addresses of the elements from memory are calculated by performing a signed-addition operation of each respective addressing value to a base address of an object in memory (Column 6, Lines 48-49, which discloses the operands would be signed).

33. As per Claim 14, Beard teaches: The method of claim 11, wherein the arithmetic-logical operation is an addition operation that produces at least one element of the result vector as a summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values (it is well known in the art that a single vector/array value can be added to multiple elements of another vector/array. For example, see Cohoon et al., Page 493, Program 10.2) corresponding to elements of the first vector register of addressing values having identical values (In correcting errors in a vector write, it would not make sense to operate on values that were already correct, therefore only the incorrect (identical/duplicate) values should be modified).

34. As per Claim 15, Beard teaches: The method of claim 11, wherein address values for the sequence of addressed locations within the constrained area of memory

are each calculated using a truncated portion of each respective addressing value of the first vector register of addressing values (Column 7 Lines 30-36. Bits 6-0 (which are truncated from the 32-bit value) are used to index into memory).

35. As per Claim 16, Beard teaches: The method of claim 15, wherein data values of the first sequence of values are each formed by concatenating a portion of each respective addressing value of the first vector register of addressing values to a respective one of a sequence of numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12).

36. As per Claim 17, Beard teaches: The method of claim 11, wherein the constrained area contains 2^N consecutive addresses, wherein address values for the sequence of addressed locations within the constrained area of memory are each calculated using an N-bit value derived from each respective addressing value of the first vector register of addressing values (Column 3 Lines 2-5 discloses adding a value to a base address, and it is well known in the art/inherent that if you have an N-bit address, it can address an area of memory of size 2^N), and wherein data values of the first sequence of values are each formed by concatenating a portion of each respective

addressing value of the first vector register of addressing values to a respective one of a consecutive sequence of integer numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12).

37. As per Claim 18, Ernst teaches: The method of claim 11, wherein for the loading of the third vector register with elements from memory, elements are loaded from locations specified by addressing values corresponding to indications that indicated compares and no elements are loaded from locations specified by addressing values corresponding to indications that indicated miscompares (Section 2.4, when selectively reloading the instructions to be replayed, only the ones that matched the criteria (i.e., not ready/valid), would have to be loaded).

38. As per Claim 19, Beard teaches: A computer-readable medium having instructions stored thereon for causing a suitably programmed information-processing system to execute the method of claim 11 (It is inherent that Beard's invention would be run off instructions stored on a computer-readable medium, otherwise it would not be possible for the computer system to execute the method).

39. As per Claim 30, Beard teaches: A system comprising:

a first vector register (Column 2, Lines 62-63, there are a plurality of registers);

a second vector register (Column 2, Lines 62-63, there are a plurality of registers);

a third vector register (Column 2, Lines 62-63, there are a plurality of registers);

a bit vector register (Column 2, Lines 62-63, there are a plurality of registers);

means for loading the first vector register with addressing values (Column 2 Line 65 – Column 3 Line 1);

means for loading the second vector register with operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

means for storing a first sequence of values to a sequence of addressed locations within a constrained area of memory, wherein each one of these location's addresses in the constrained area of memory is based at least in part on a subset of bits of a corresponding one of the addressing values (Column 3, Lines 19-22);

means for loading from the sequence of addressed locations values resulting from the storing of the first sequence to obtain a second sequence of values (Column 3 Lines 5-10);

means for selectively combining, with an arithmetic-logical operation, certain elements of the second vector of operand values based on results of the comparing (Column 3, Lines 10-13);

means for performing the arithmetic-logical operation using values from the third vector register and the combined second vector of operand values to generate a result vector (Column 3, Lines 10-13); and

means for storing the result vector to memory (Column 3, Lines 11-13), but fails to teach:

means for comparing the first sequence of values to the second sequence of values;

means for loading a third vector register with elements from memory address locations generated using at least some of the first vector register of addressing values.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

40. As per Claim 31, Beard teaches: A system comprising:

a first vector register that can be loaded with addressing values (Column 2 Line 65 – Column 3 Line 1) ;

a second vector register that can be loaded with operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

a third vector register that can be loaded with operand values from memory locations indirectly addressed using the addressing values from the first vector register (Column 3, Lines 19-22, while Column 28, Lines 18-19 show indirect addressing capabilities);

a circuit that selectively adds certain elements of the second vector of operand values based on the element addresses the duplicated values (Column 3, Lines 10-13, where addition is a common operation that can be used);

a circuit that uses indirect addressing to selectively load the third vector register with elements from memory (Column 3 Lines 5-10, while Column 28, Lines 18-19 show indirect addressing capabilities);

a circuit that selectively adds values from the third vector register and the second vector of operand values to generate a result vector (Column 3, Lines 10-13); and

a circuit that selectively stores the result vector to memory using indirect addressing (Column 3, Lines 11-13, while Column 28, Lines 18-19 show indirect addressing capabilities), but fails to teach:

a circuit that determines element addresses of the first vector register that have a value that duplicates a value in another element address.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57).

One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

41. As per Claim 32, Beard teaches: The system of claim 31, further comprising:

an adder that generates addresses of the elements from memory by adding each respective addressing value to a base address (Column 3 Lines 2-5).

42. As per Claim 33, Beard teaches: The system of claim 31, further comprising:

an adder that generates addresses of the elements from memory by a signed-addition operation of each respective addressing value to a base address of an object in memory (Column 6, Lines 48-49, which discloses the operands would be signed).

43. As per Claim 34, Beard teaches: The system of claim 31, wherein the circuit that selectively adds certain elements performs one or more addition operations using those values from a plurality of respective elements of the original second vector of operand values (it is well known in the art that a single vector/array value can be added to multiple elements of another vector/array. For example, see Cohoon et al., Page 493, Program 10.2) corresponding to elements of the first vector register of addressing values having identical values (In correcting errors in a vector write, it would not make

sense to operate on values that were already correct, therefore only the incorrect (identical/duplicate) values should be modified).

Conclusion

44. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure as follows. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

45. Koyanagi (USPN 6,816,960) teaches a machine with vector-scatter and vector-gather operations, with a collision/coincidence detection unit.

46. Hall et al. (USPN 5,418,916) teaches re-executing failed instructions in a vector processor.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Robert E Fennema
Examiner
Art Unit 2183

RF

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100